

## **Perl e Win32: installazione e utilizzo**

- Introduzione**
- Disponibilità**
- ActivePerl: Installazione**
- Moduli: Installazione tramite PPM**
- Perl e IIS**
- Link**

## Introduzione:

Perl, acronimo di Pratical Extraction and Report Language subito ribattezzato Pathologically Eclectic Rubbish Lister dai primi sviluppatori, è un linguaggio interpretato originariamente sviluppato da Larry Wall, autore di famosi programmi Unix come patch(1) e rn(1), e distribuito tramite Usenet.

Perl, nato in ambiente Unix, riassume funzionalità tipiche di programmi come awk e sed(1), nonché caratteristiche tipiche della programmazione Shell e un interfaccia per le chiamate di sistema a primitive Unix.

Il linguaggio Perl è un **interpretato**, anche se funziona in maniera leggermente diversa dalla maggior parte dei linguaggi che posso essere accumulati da questa definizione, infatti Perl svolge contemporaneamente funzionalità di compilatore e interprete. Quando un file sorgente Perl viene dato in pasto all'interprete, tale file viene prima compilato e ottimizzato e quindi eseguito.

La compilazione della sorgente permette anche di riscontrare eventuali errori di sintassi in quanto l'esecuzione non avviene fino a che il programma non sia stato compilato per intero.

Questo permette non solo di ottenere un'esecuzione a livello runtime efficiente ma anche di migliorare il ciclo di sviluppo del software in quanto è possibile apportare modifiche allo script e testarle immediatamente senza dovere incorrere in cicli spesso lunghi e noiosi di compilazione e linkaggio.

Perl per sistemi Win32 vide la luce quando Microsoft diede incarico a ActiveState Tool Corporation (derivata da Hip Communication) di sviluppare una migrazione dell'interprete in modo da poterlo includere all'interno del Windows NT Resource Kit.

ActiveState ( [www.activestate.com](http://www.activestate.com)) si occupa tuttora di migliorare e distribuire tale interprete, nonché estenderne le funzionalità specifiche Win32 e includervi nuove funzionalità mano a mano che vengono proposte nella distribuzione originale.

Perl per Win32 consente di utilizzare la maggior parte degli strumenti disponibili ai programmatori in ambiente Windows come OLE automation, connessione con database tramite ODBC, sviluppo ActiveX e molto altro.

Anche se l'utilizzo principale di questo linguaggio è oggi indirizzato verso applicazioni Web di tipo CGI esso può anche essere utilizzato per produrre estensioni ISAPI (funzionalità aggiuntive a quelle offerte dal Web Server) o anche come linguaggio di scripting ActiveX.

Utilizzando le funzionalità offerte dal riconoscimento e manipolazione di **Espressioni Regolari** (RE) è possibile costruire tool per la validazione sintattica di file HTML, tramite moduli specifici è possibile modificare il Registro di Sistema (Registry), l'Event Log e addirittura le informazioni relative agli utenti WinNT.

Utilizzando moduli, liberamente scaricabili, come Win32::ODBC o Active Data Objects (ADO) è possibile interagire con database sia locali che remoti.

Tramite script Perl è possibile scaricare e filtrare la propria posta o messaggi da Newsgroup, inviare e-mail, interagire con server HTTP e FTP e creare client per qualsiasi altro tipo di server Internet.

**Disponibilità:**

Perl è distribuito sotto licenza pubblica GNU, ciò significa in poche parole che:

“Puoi distribuire i file binari del Perl solo se rendi disponibili gratuitamente anche i sorgenti, se modifichi il Perl devi rendere disponibile anche i sorgenti modificati”.

La licenza “Artistic License”, che accompagna i sorgenti, rende comunque legittimo il vendere applicazioni proprietarie sviluppate in Perl.

La distribuzione di Perl per Win32 della ActiveState è disponibile sul sito [www.activestate.com](http://www.activestate.com), è possibile inoltre trovare i sorgenti e i compilati (binari) sul sito della CPAN: [www.cpan.org](http://www.cpan.org), la quale fornisce inoltre file precompilati per le piattaforme più utilizzate (IRIX, Alpha, Macintosh, ...).

Per iniziare è consigliabile fare riferimento alla distribuzione ActiveState che è rilasciata come unico file eseguibile auto-installante.

Tale distribuzione comprende:

- **Perl per Win32:** distribuzione binaria dell'interprete Perl
- **Perl per ISAPI:** plug-in per Internet Information Server
- **PerlScript:** un motore per lo scripting ActiveX
- **Perl Package Manager (PPM):** utility per l'installazione e la gestione dei moduli Perl

I requisiti hardware minimi per l'installazione sono:

Processore Intel 80486, raccomandati Alpha o Pentium, almeno 16MB RAM per Windows NT, 32 MB consigliati.

8MB RAM per Windows 95, 16MB consigliati.

Almeno 20MB di spazio su disco per un'installazione tipica.

Sistemi Operativi:

Windows NT 4.0 (service pack 3 raccomandato) , Windows 98 o Windows 95.

Perl per ISAPI:

Un Web Server che supporti le estensioni ISAPI come ad esempio IIS 3.0 o maggiori.

PerlScript:

Un host che supporti gli script basati su ActiveX come Internet Explore 4.0 o maggiori, IIS 3.0 o IIS 4.0.

**Installazione:**

Eseguendo il pacchetto auto-installante fornito da ActiveState (ad esempio APi522e.exe), si presenterà un classico wizard InstallShield per l'installazione, tramite il quale sarà possibile decidere **cosa** e **dove** installare.

La directory di default proposta è **C:\Perl**, nel caso più comune in cui il sistema sia installato sul driver C.

Le modifiche apportate al sistema saranno le seguenti:

- la DLL (Dynamic Linked Library) **PerlCRT.dll** sarà installata nella directory di sistema: in genere **C:\WINNT\System32** per sistemi WinNT e **C:\WINDOWS\System** per Win95/98
- vengono create nel registro di sistema le chiavi:

```
\\HKEY_LOCAL_MACHINE\Software\ActiveState\ActivePerl\<build #>  
\\HKEY_LOCAL_MACHINE\Software\Perl
```

E' consigliato, quando possibile, utilizzare una macchina WinNT in quanto alcuni dei moduli specifici per Win32 non funzionano con Win95, inoltre la scelta di un WinNT Server sembra obbligata se si voglia utilizzare IIS 4.0. Nel caso non si disponga di adeguate licenze è sempre possibile optare per un Web Server freeware come ad esempio il conosciutissimo Apache in versione Win32, reperibile all'URL:

<http://www.apache.org/docs/windows.html>

A questo punto dopo avere controllato che la variabile di environment PATH contenga il percorso corretto dell'eseguibile perl, per default **C:\perl\bin\**, è possibile eseguire uno script da linea di comando ad esempio:

```
perl \perl\eg\example.pl
```

Per sistemi WinNT associando l'estensione **.pl** all'interprete sarà possibile lanciare lo script direttamente scrivendo il nome dello stesso:

```
\perl\eg\example.pl
```

## Moduli: Installazione tramite PPM

Uno dei punti di forza del linguaggio Perl è dato dall'enorme numero di persone che collaborano attivamente alla realizzazione, mantenimento e distribuzione di **moduli (package)** che offrono **metodi** e **funzioni** che permettono di scrivere facilmente script per rispondere alle più diverse esigenze applicative.

ActiveState si occupa, tra l'altro, di sviluppare moduli ad hoc e di rendere compatibili quelli già esistenti per applicazioni in ambiente Win32.

I moduli distribuiti da ActiveState possono essere scaricati in formato .zip all'URL:

<http://www.activeState.com/packages/zips/>

L'installazione standard del Perl per Win32 fornisce anche un utile strumento per la gestione dei moduli: **PPM** (Perl Package Manager).

PPM fornisce, tramite un'interfaccia da linea di comando, la possibilità di installare, rimuovere o effettuare l'upgrade di moduli Perl.

Per installare package non presenti sul proprio hard disk è necessario innanzitutto essere connessi ad Internet, se la connessione avviene tramite un server proxy si dovrà settare la variabile di environment HTTP\_proxy.

A questo punto si può lanciare il programma digitando

**ppm**

da linea di comando (MS\_DOS Prompt). Tale comando si riferisce al file

ppm.bat

che si trova per default nella directory

C:\perl\bin

Si entrerà così nella shell interattiva fornita dal programma, riconoscibile dal fatto che il consueto prompt della finestra DOS sarà PPM>

Il primo approccio con il Gestore di Moduli può essere il digitare

**search**

che visualizza una lista dei package resi disponibili da ActiveState, per installarne uno è sufficiente eseguire il comando

**install nome\_modulo**

Per vedere i moduli già installati sul proprio sistema, ovviamente in questo caso non è necessario essere connessi, utilizzare il comando

**query**

Per vedere i comandi disponibili:

**help**

che associato ad un particolare comando ne mostra l'utilizzo esempio:

**help genconfig**

## Perl e IIS

Su una macchina WinNT con IIS 3.0 o maggiore, l'installazione del Perl della ActiveState "dovrebbe" provvedere ad associare l'interprete ai file con estensione .pl e la dll perlis.dll a quelli con estensione .plx.

In genere è bene controllare che tali modifiche siano state apportate e in caso negativo provvedere "a mano":

-PANNELLO di CONTROLLO → SISTEMA → AMBIENTE → verificare che la variabile **PATH** contenga la directory dell'interprete.

-utilizzando Microsoft Management Console se si associa un applicazione alla directory preposta a contenere le CGI (cgi-bin) è necessario impostare in APPLICATION SETTINGS → CONFIGURATION all'estensione .pl il comando **c:\perl\bin\perl.exe "%s" "%s"** sempre che il path scelto per l'installazione sia quello di default.

A questo punto riavviare il Web Server.

In ambiente Win32 e IIS il concetto di CGI è contrapposto a quello di ISAPI.

CGI (Common Gateway Interface) è un modo per estendere le funzionalità di un server permettendo al server di avviare (spawn) e interagire con un programma eseguibile o script "esterno" il cui output viene inviato al client: il browser dell'utente.

ISAPI che è uno standard specifico dei server Microsoft non esiste il concetto di "avvio" (fork) di eseguibili separati dal processo padre, invece c'è una dll (nel nostro caso perlis.dll) caricata nello spazio di memoria del Web server all'avvio dello stesso.

In questo modo quando arriva una richiesta per l'interpretazione di uno script Perl, non c'è bisogno di generare un nuovo processo che avrebbe costi, a livello di prestazioni, molto alti per WinNT ma è la dll che si prende carico dell'esecuzione dello script.

Un problema, come è facile intuire, è dato dal fatto che essendo la memoria condivisa tra la dll e il server, se lo script compie azioni illecite (accessi ad aree di memoria non consentiti, etc.) rischia di bloccare anche il server, questo è uno dei motivi per cui il tempo medio fra due riavvii è molto minore per IIS su WinNT che ad esempio Apache su Linux.

## **Link**

### **download:**

[www.activestate.com](http://www.activestate.com)

[www.cpan.org](http://www.cpan.org)

### **documentazione:**

[www.microsoft.com](http://www.microsoft.com) ☺

[www.15seconds.com](http://www.15seconds.com)

[www.perl.com](http://www.perl.com)

[www.perl.org](http://www.perl.org)

[perl.oreilly.com](http://perl.oreilly.com)

[www.troubleshooters.com/codecorn/littperl/](http://www.troubleshooters.com/codecorn/littperl/)

Mario Protto

[mario@mario-online.com](mailto:mario@mario-online.com)

[www.mario-online.com](http://www.mario-online.com)